WE CLAIM:

1. A method of optimizing the instruction set of a digital processor, comprising:

(i) providing a program having a plurality of different instruction types;

(ii) determining the frequency of each of said instruction types from a base instruction set;

(iii) determining the number and type of instructions necessary for correct instruction set execution based at least in part on said act of determining the frequency; and

(iv) creating a compressed instruction set encoding to generate a compressed instruction set based at least in part on said act of determining.

2. The method of Claim 1, further comprising:

re-evaluating said compressed instruction set using at least said steps (i), (ii), and (iii); and

generating an instruction set encoding for said compressed instruction set.

3. The method of Claim 1, wherein the act of providing a program comprises providing an assembly language program.

4. The method of Claim 3, further comprising sorting said instruction types by frequency of usage.

5. The method of Claim 4, wherein said digital processor includes an extension logic unit adapted to execute at least one extension instruction, and the act of providing comprises providing a program having said at least one extension instruction, aid at least one execution instruction being executable by said extension logic unit.

6. The method of Claim 1, wherein the act of creating a compressed instruction set comprises selecting those "N" instructions having the greatest frequency of occurrence, said selected "N" instructions permitting said program to be compiled with a predetermined size.

7. The method of Claim 6, further comprising the act of determining a compression ratio for said compressed instruction set, said compression ratio being

related to the ratio of the number of compressed instructions to the total number of original instructions.

8. A pipelined digital processor, comprising:

a processor core having an instruction pipeline comprising at least instruction fetch, decode, and execute stages;

a data interface in data communication with said processor core, said interface adapted for data communication with a storage device configured to hold a plurality of program instructions; and

an optimized instruction set comprising a base instruction set and a compressed instruction set, said compressed instruction set being generated by the method comprising:

determining the frequency of each of said instruction types from said base instruction set;

determining the number and type of instructions necessary for instruction set execution based at least in part on said act of determining the frequency; and

creating a compressed instruction set encoding to generate said compressed instruction set.

9. The processor of Claim 8, wherein the act of creating a compressed instruction set comprises selecting those "N" instructions having the greatest frequency of occurrence, said selected "N" instructions permitting said program to be compiled with a predetermined size.

10. The processor of Claim 8, wherein said optimized instruction set also comprises at least one extension instruction adapted to perform a predetermined function, said processor further comprises an extension logic unit adapted to execute said at least one extension instruction.

11. The processor of Claim 9, further comprising an encoding structure having an opcode and a plurality of instruction slots.

12. The processor of Claim 11, wherein said plurality of instruction slots comprise two slots, each of said slots having two 14-bit instructions.

13. The processor of Claim 11, wherein said encoding structure comprises 32 bits, and said opcode is disposed within the last four bits thereof.

14. A digital processor, comprising:

a processor core having a pipeline comprising at least instruction fetch, decode, and execute stages;

a memory interface adapted to at least read program instructions from a program memory and provide said instructions to said pipeline; and

an optimized instruction set comprising a base instruction set and a compressed instruction set, said compressed instruction set having a predetermined number and type of instructions necessary for correct instruction set execution on said processor core, said predetermined number and type based at least in part on the frequency of occurrence of instructions within said base instruction set.

15. The processor of Claim 14, further comprising an encoding structure having an opcode and a plurality of instruction slots.

16. The processor of Claim 15, wherein said instruction set includes at least one extension instruction, said at least one extension instruction adapted to perform a predetermined function upon execution within said processor.

17. The processor of Claim 16, further comprising an extension logic unit adapted to execute said at least one extension instruction.

18. A method of synthesizing the design of an integrated circuit digital processor comprising:

obtaining input regarding the configuration of said design;

creating a functional description based on said input and existing libraries of functions, said functional description comprising an optimized instruction set comprising a base instruction set and a compressed instruction set, said compressed instruction set having a predetermined number and type of instructions necessary for correct instruction set execution on said processor, said predetermined number and type based at least in part on the frequency of occurrence of instructions within said base instruction set;

determining a design hierarchy based on said input and existing libraries;

generating structural HDL and a script associated therewith;

running said script to create a synthesis script; and

synthesizing said design using synthesis script.

19. A method of enhancing the performance of a reduced instruction set processor, said processor having a multi-stage instruction pipeline and an instruction set having at least a base instruction set, said base instruction set having a plurality of instruction types associated therewith; comprising:

providing a program having a plurality of instructions;

determining the frequency of each of said instruction types within said plurality of instructions of said program;

selecting those "N" instructions having the greatest frequency of occurrence, said selected "N" instructions allowing said program to be compiled with a predetermined size; and

compiling said program based at least in part on said "N" instructions.

20. An application specific integrated circuit (ASIC), comprising:

a first processor core, said processor core having a pipeline with at least instruction fetch, decode, and execute stages associated therewith;

an optimized instruction set comprising a base instruction set and a compressed instruction set, said compressed instruction set having a predetermined number and type of instructions, said predetermined number and type based at least in part on the frequency of occurrence of instructions within said base instruction set, said optimized instruction set further comprising at least one extension instruction adapted to perform at least one specific operation;

at least one storage device adapted to store a plurality of data bytes therein, said at least one storage device being accessible by said first processor core; and

at least one extension logic unit adapted to facilitate execution of said at least one execution instruction.

21. The ASIC of Claim 20, further comprising a second processor core, said second processor core being disposed on the same die as said first processor core.

22. The ASIC of Claim 21, wherein said second processor core comprises a digital signal processor (DSP), said DSP being adapted to perform at least one operation on data provided to said ASIC.

23. The ASIC of Claim 22, wherein said DSP is adapted for initiation by an instruction from said first processor core.

24. The ASIC of Claim 22, wherein at least a portion of the operation of said DSP is controlled by extension registers associated with said first processor core.

25. A pipelined digital processor, comprising:

processor means having an instruction pipeline comprising at least means for instruction fetch, means for instruction decode, and means for instruction execution;

means for data interface, said means for data interface being in data communication with said processor core, said means for data interface adapted for data communication with a storage device configured to hold a plurality of program instructions; and

optimized instruction means comprising base instruction means and compressed instruction means, said compressed instruction means being generated by the method comprising:

determining the frequency of each of said instruction types from said base instruction means;

determining the number and type of instructions necessary for instruction execution based at least in part on said act of determining the frequency; and

creating a compressed instruction set encoding to generate said compressed instruction means.

26. A method of operating a pipelined digital processor having an instruction pipeline comprising at least instruction fetch, decode, and execute stages, a storage device configured to hold a plurality of program instructions, and an optimized instruction set, the method comprising:

providing a base instruction set having a plurality of instructions;

providing a compressed instruction set derived at least in part from said base instruction set;

assigning one of a plurality of predetermined values to at least one bit within a status register within said processor;

executing at least one instruction from said base instruction set within said pipeline based on a first predetermined value present in said status register; and

executing at least one instruction from said compressed instruction set within said pipeline based on a second predetermined value present in said status register.

27. The method of Claim 26, wherein the act of assigning comprises assigning a "1" or "0" value to a low address (L) bit within said register.

28. The method of Claim 27, wherein the act of providing a compressed instruction set comprises encoding at least a portion of the instruction operation codes (opcodes) for the compressed instructions within the "n" most significant bits.

29. The method of Claim 27, wherein the act of providing a compressed instruction set further comprises encoding a plurality of said compressed instructions with source register fields located in a predetermined relationship to one another.

30. The method of Claim 29, wherein the act of encoding with said predetermined relationship comprises encoding the source register fields for respective ones of said plurality of compressed instructions at identical locations.

31. The method of Claim 26, wherein the act of providing a compressed instruction set comprises encoding all of the immediate data fields such that they start from the least significant bit (LSB).